NoSQL Proof-of-Concept Proposal – Dota2 Game Replay Analysis

Team name: PSG.LGD

Yichun Yan
Ziwei Jiang
Yifan Li
Weiqi Wang

October 4, 2019

Contents

1   Project Background

Our team aims at exploring data about a popular and long-lived computer game, Dota2. Dota2 is a multiplayer online battle arena (MOBA) video game developed and published by Valve Corporation. Each game will involves ten players that are divided to two teams play against each other on a same map. Every player chooses one "hero" that has unique abilities and different styles of playing. The aim for each team is to destroy the building called "Ancient" of the other team. Players collect gold to buy powerful items and gain XP by killing players on the other team, destroy building of the other team, or kill creeps.

Figure 1: A screenshot of the game

This game has a large quantity of players, accordingly a huge amount of replays will be produced every day. Also, it's a real-time game, hundreds of operations and some decisions will be made each second, and all of them will have some influence on the final results of the game. Both of the above two features make the game a great subject to big data analysis. Actually, there are some professional teams start hiring data engineers to help their players perform better in a game. This means that our analysis has some realistic meaning.

Knowing that this game is not familiar to everybody, and we may use some words that may be difficult to understand to those who don't play this kind of games for simplicity. I create a glossary at the end of this proposal for your reference.

2   Potential Datasets

We will mainly use two datasets:

1. Dota2 match result dataset
2. Dota2 replay dataset

The first dataset is the snapshot of the final state of each player and the whole game, containing only important information. The second one is a binary format file which can be executed by Dota2's client to reproduce everything happened in the game.

The datasets can be retrieved from the following resources:

1. Valve's Dota2 replay servers
2. Valve's official API

We will get the match results data from Valve's official API. It also provides a key, which we can use to retrieve a replay from the replay server.

In all, we will have two different kinds of data for our further analysis, relatively large-scale but coarse-grained match results data, and very fine-grained replays data. Considering the file size of one replay, we decide to include recent professional games and randomly select some public games for the replay data. We will mention how we acquire the data in Data ETL section.

3 Five Vs of Datasets

To assess the adequacy of adopting big data to our project, we firstly identify five Vs characteristics of our dataset.

3.1 Volume

Replays are one of the datasets we plan to use, which are the full records of finished matches. The size of one replay is estimated to be about 60MB. We plan to collect 3000 replays whose size is about 180GB. As for match result data which is JSON-format data, the size of each result is about 5KB. We plan to collect 10,000,000 match results whose size is about 50GB. Moreover, if we desire more data in future implementation, we can always access it as long as Dota2 still has players.

3.2 Velocity

According to the statistic data in August 2019 from Steam which is a video game digital distribution platform, the average number of players is 467,148.3 players and the peak number is 826,690. A large number of players makes both the replay data and result data high-velocity. Although in our project, the velocity depends on the velocity of the API request, this statistic data reveals that it's potential to be high-velocity.

3.3 Variety

From the APIs mentioned above, we can access JSON format files, the match result data, and binary format files, the replay data.

3.4 Veracity

The overall quality of our datasets is good because we access most of them from the official API. But there is still a few noises. For example, the time of some games is too short, which makes these game lack of representativeness. Thus, we plan to implement data validation to filter the undesired matches.

3.5 Value

Dota2 game statistics and replays data are valuable. Professional teams may use it to analyze the performance of team members or design new strategies for competitions. Amateur players can also learn a lot about the techniques, timing, hero choice from the data. Our business questions, will focus on these aspects to provide valuable insights.

Even though the game is versioning, our datasets can be useful for a relatively long time. Because we plan to collect 1/3 professional matches and 1/3 public matches, 1/3 ranked matches. The professional matches are durable since we can always extract information about professional teams and players from them. As for the public games, they are valuable as long as they are not too stale. Minor changes of the game does not affect the whole picture. But we will filter out public games which are too outdated in the data validation step, in order to make our datasets more valuable.

4   Potential Business Questions

As mentioned before, we can collect 2 kinds of data, one is match results and one is replays. By analyzing them, the following questions might be answered.

- Easy questions:
    - Who is the hero gaining gold/XP fastest in 15 minutes in public games/professional games?
    - Who is the hero having most kills/assists/heals/deaths in public matches/ranked matches/professional matches?
    - What is the most purchased item in public matches/ranked matches/professional matches?
    - Who is the hero having most bad-manner players (players who are AFK or disconnected)?
    - Who is the most popular hero (the hero who has the highest pick rate) in public matches/ranked matches/professional matches?
    - Who is the hero having the highest ban rate in professional matches?
    - who is the hero having the most ban/pick rate in professional matches?
    - How long does a game cost on average in public matches/ranked matches/professional matches?
    - Which item is used most in public matches/ranked matches/professional matches?

We categorize these questions as easy questions because the solutions to them are straight-forward. These questions can be answered by implementing aggregation on our datasets. And most of the data needed for these questions can be extracted directly from our datasets.

- Moderately challenging questions:
    - How are the benefits gained from buybacks in public matches/ranked matches/professional matches□
    - How is the vision in public matches/ranked matches/professional matches?
    - When does the first team battle happen in public matches/ranked matches/professional matches?
    - Which is lineup having the highest win rate in public matches/ranked matches/professional matches?
    - Which is the most popular lineup (lineup which has the highest pick rate) in public matches/ranked matches/professional matches?
    - Which is the most common ban-pick combo in public matches/ranked matches/professional matches?
    - Who is the hero changed most in win/pick/ban rate after releasing a new version of Dota2?
    - Who is the hero having the highest winning rate in each lane?

We categorize these questions as moderately challenging questions because the solutions to them are not that straight-forward as easy questions. The first challenge is to define some terms we used in these questions in the application code. For example, we use the term, vision, which is involved with the lifecycle of ward and sentry (two items provide visibility around them). Besides aggregation, we also need join operations to answer these questions.

- Challenging questions:
    - How is the winning-losing relationship between Dota2 professional teams? Can we use a graph to visualize it?
    - Does there exist a regular farming path for some professional players?

– Does there exist a correlation between the time of the first-blood and the time of the entire game?
– Does there exist a correlation between the gold/XP source of a hero and its win rate?
– Does there exist a correlation between the distribution of the economy and the result of the game?
– Does there exist a correlation between economic development and the time of the first team battle?

We categorize these questions as challenging questions because they need more techniques than moderately challenging questions. Like moderately challenging questions, they have some ambiguous definition needs to be implemented. But they cannot be answered easily by aggregation and join operations. Instead, we plan to use machine learning techniques to solve them.

5 Data ETL

5.1 Extraction

Valve, the company that develops the game, initially try to provide all the players with easy APIs to access the data of the game. But due to the increasing stress on its data servers, many of its APIs are shut down. At the same time, the documentation seems to be never updated since it's created, which makes it much more difficult for us to collect the data. For example, the API which returns a bunch of the match results given a starting match id is not usable anymore. Another API which returns a key that is critical for construct the URL to download replay of that game is not working as well.

But luckily, we found some hints from the developer's forum of Dota2. We can use the starting sequence number, which works similarly as the match id, to get a series of game results. And by calling a third party API from OpenDota, we can get the important key to constructing the URL to download replays again.

The specific steps of gaining our data are as following:

1. Match Results:
   - Use GetMatchHistoryBySequenceNum API to get the match ids. We will need a a field called `start_at_match_seq_num` to specify the starting match sequence number of the results.
   - Then we can extract the last sequence number of the results as the `start_at_match_seq_num` for the next call. By doing this iteratively we can enlarge our dataset for our first kind of data.

2. Replays:
   - Get the match ids of recent professional games from the last step.
   - Use the those ids via OpenDota API to get the information we need (`cluster` and `replay_salt`) for retrieving replays from Valve's replay server
   - Construct links in this format: `http://replay<cluster>.valve.net/570/<match_id>_<replay_salt>.dem.bz2` to get the replays.

5.2 Transformation

The match result is in JSON format so we can almost directly store it. The information it contains is:

- Players information:
  - Account id
  - Player slot
  - The hero he uses in this match
  - The items he possesses at the end of the match
  - The player's kills/deaths/assists
  - The player's leaver status (whether AFK/disconnected or not)
  - The gold he possesses at the end of the match
  - The amount of last-hits and denies the player got during the match

     – The average gold/XP he gained per minute
     – The total gold he spent during the match
     – The amount of damage the player dealt to heroes/towers
     – The amount of health the player had healed on heroes
     – The player's level at match end
     – A list detailing a player's ability upgrades, including the ability and the upgrading time
     – Additional playable units owned by the player and its items

- The season the game was played in
- The winner team of the match
- The length of the match
- Unix timestamp of when the match began
- The matches unique ID
- A sequence number, representing the order in which matches were recorded
- The tower status of both teams
- The barracks status of both teams
- The server cluster the match was played upon
- The time in seconds since the match began when first-blood occurred
- The mode of the match
- The number of human players within the match
- The league that this match was a part of
- The number of thumbs-up/thumbs-down the game has received by users
- A list of the picks and bans in the match, if the game mode is Captains Mode

On the other hand, the replay data is completely unstructured `.dem.bz2` binary file, data transformation must be done before we can store the data in our database.

Firstly, we can decompress the `.bz2` file with `org.apache.commons.compress.compressors.bzip2` package in Java, which will give us a `.dem` file:

Next, we can utilize clarity, an open source Dota2 replay parser, to extract useful information from the `.dem` file.

It cannot be achieved by a single click, though. We have to write a lot of code to invoke its function. What's more, clarity does not provide detailed documentation, instead there are only some examples which forces us to iteratively attempt and learn the usage of this tool.

Based on our exploration, the following data will be available:

- Player name, id, team formation and hero choice
- Detailed log of the game, including a hero:
     – deals damage to another one
     – heals another one
     – receives/loses a buff/debuff
     – kills another one
     – uses his ability
     – uses an item
     – buys an item
     – receives/loses some gold
     – gains some XP
     – buys back (spending money in order to instantly re-spawn)
- Spawn/death of heros and NPCs

These information will be organized in a Document and store to our database.

Most fields in our datasets are related to some of the above-mentioned business questions. They are all

important. We will generally validate them, for example, fields like the HP, XP, total gold cannot be negative.

What's more important is to filter out invalid data on the game level. Some of the games lasts for only two or three minutes, because some players are disconnected and others just quit the game very quick. Others are practicing games where most players are computer bots. We will completely drop these game records.

5.3   Loading

5.3.1   NoSQL Storage Technology

As we mentioned above, we have two datasets - one is the match results data in JSON format, which is a kind of document that can be encoded using a text-based encoding scheme. The replay data is binary files which is inappropriate for document NoSQL database to store. But after we parsing those replay data, we can gather useful information from replay data in Document format.

Therefore, we choose to use MongoDB as our primary NoSQL storage technology.

MongoDB is a popular NoSQL storage device that can store high volume, high velocity, high variety Big Data datasets. It's a document storage device, which makes storing semi-structured document-oriented data such as JSON much easier. It is an open-source NoSQL database, so we are free to use it.

Also, MongoDB supports partial update. This will help us aggregate values in future games that we would potentially use.

To answer the business question about the relationship between professional teams and players, using only MongoDB may involve considerable aggregations. We may consider using a graph database, neo4j, as our secondary NoSQL technology choice.

5.3.2   Docker

In order to combine different needs, programming languages and developing environment of many technologies, we will adopt Docker to our project. Using Docker, we can deploy our application into any environment.

Docker is a container technology that provides consistency. It's open-source and there are many open source applications that have been made in Docker Hub. Docker Hub provides official or verified images, so it's safe to use.

In our project, we will build Dockerfile to let it connect with our project. We will use Scala as programming language, and we will get the match results data from Dota2's official API. MongoDB will be used as database. And then we will put all of this into a Docker container.

5.3.3   AWS (Amazon Web Service)

As we mentioned before, we need 50GB to store the JSON file, and we need 180GB to store replay data, and the scale would be extended if we add more data in the datasets. In order to have enough capacity to store datasets, we will use AWS (Amazon Web Server) to deploy our project on a cloud server. Running Docker on AWS will provide us an open-source and free way to build and run a distributed application and provide a reliable and easy way to scale out.

AWS provides numbers of official ways to deploy docker on AWS, such as Amazon ECS, which is a highly scalable, high-performance container orchestration service to run Docker containers on the AWS cloud. After we package datasets and analytics packages into Docker, we will deploy Docker to Amazon ECS.
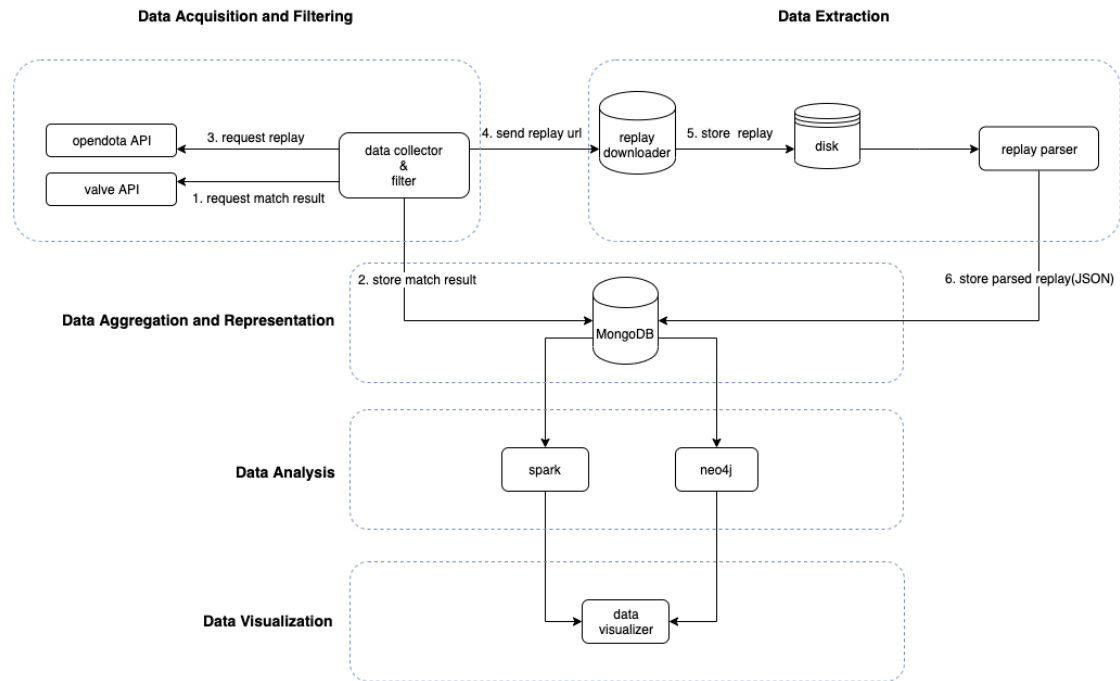
6 Overview of our Architecture

Figure 2: Our Project's Structure

7 Potential Challenges

There are several aspects where we may have potential challenges.

During the environment setup:

- We are trying to build a docker containerized application, but none of our team members have previous experience
- We do not have experience about using AWS or other cloud services, neither

In the data ETL phase:

- Download replays from the server can be slow, the links may not work
- Professional games are relatively rare, so more efforts will be needed to find and retrieve the professional replays
- Both APIs have call limits, even though we can control our request frequency, some request may still fail
- Some downloaded replay files may be broken so that no information can be extracted, or they may even cause exceptions in the pipeline
- The open-source replay parser does not have documentation, thus some of our parsing code may cause unexpected exceptions when there are some corner case
- Although we try our best to write the code to handle errors to avoid the program failing, but there still can be cases that are not covered, since our architecture involves a lot of network communications and i/o. We might need to design a log system to help us restore from a crash.

In the data analysis phase:

- Lots of joins and aggregations between the two datasets will be involved
- We must quantitatively define some generally used concepts which are often qualitatively used, for example, the measurement of vision, team battle, etc. An inappropriate definition can make the whole conclusion meaningless
- When we are answering the relationship between the teams or players, we will have to extract and load the data to a secondary database, like neo4j

Glossary